CHEF

THE PAINS, THE GROWTH AND THE UGLY.

THE UGLY

Cassiano Aquino
- Tech Lead DevOps Engineer
- Zendesk
- http://syshero.org/
- http://github.com/caquino
- @syshero
- Linux user since 1995
- Chef user since 2011

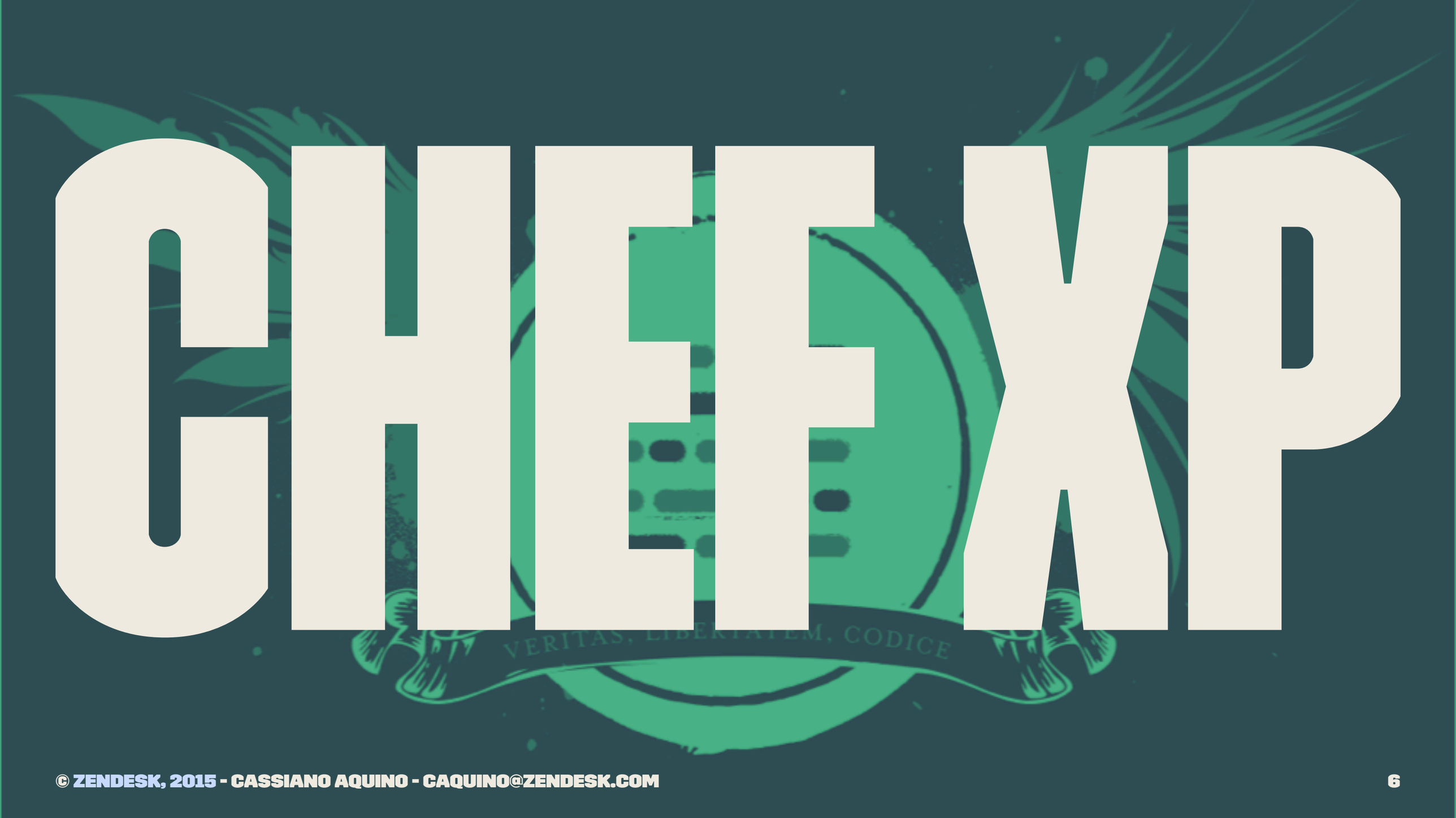
Wrap-up and the presentation will be available in the next few days on my blog.

CHEF @ ZENDESK

CHEF @ ZENDESK

Enterprise Chef 12

- 181 roles
- 200 cookbooks
- ~2000 nodes
- ~93,600 convergences day
- ~28,000,000 attributes

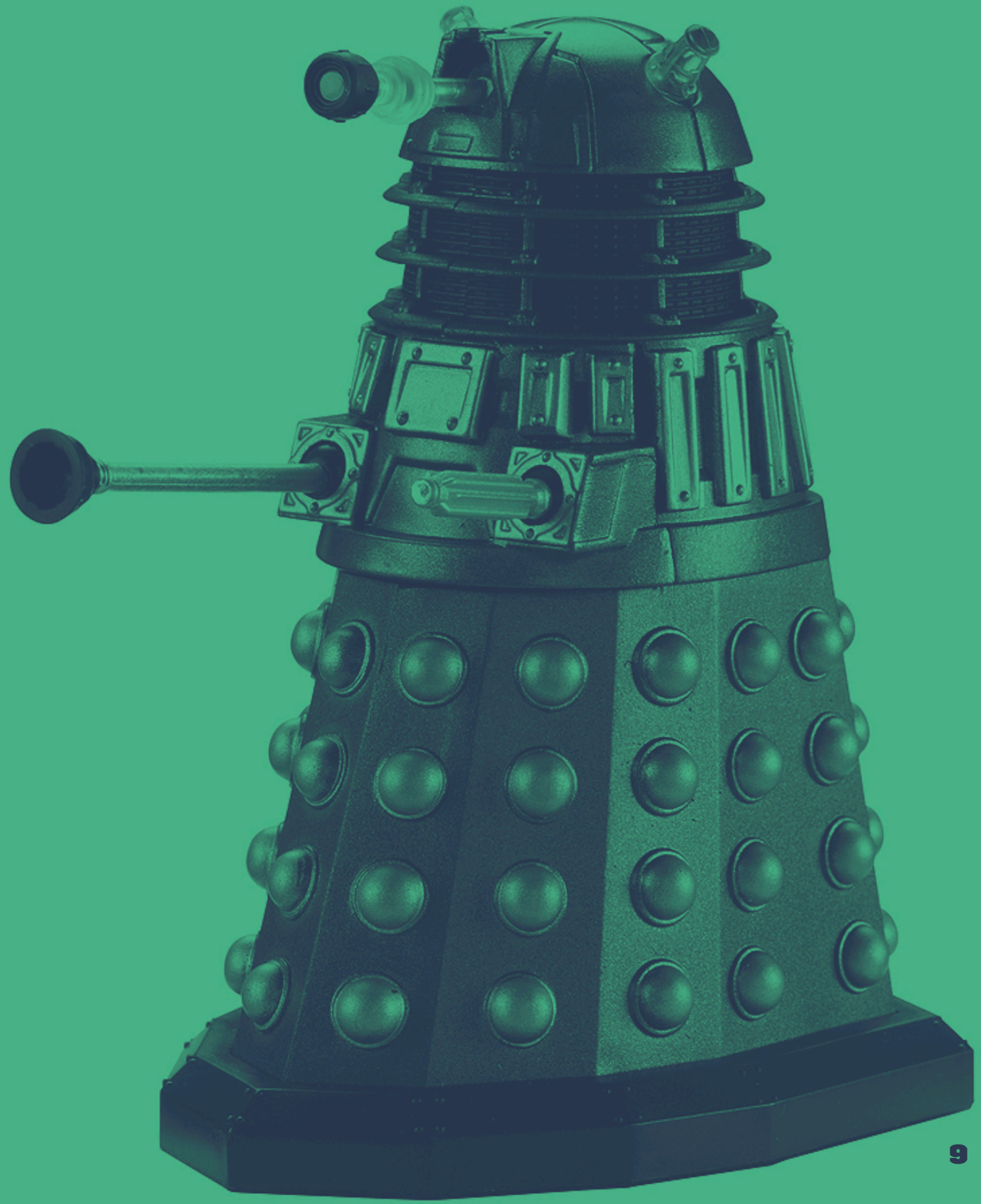


BEGINNING PARTY

Everything works
- Attributes, a lot of attributes!
- Fast convergences
- Search everywhere

AFTER PARTY HANGOVER

Why it's not working anymore?
- Attribute hoarder
- Run times bigger than interval
- Search? partial search!
- DDoS
- Exterminate! Exterminate!



DDOS?

ALMOST!

avg(node-size) * conv/d * nodes = transferred data
- avg(node-size): 150KBytes
- Convergences day: 288 (1 node every 5 minutes)
- Nodes: 2000

150 * 288 * 2000 = 86,400,000 (~ 82 GBytes)

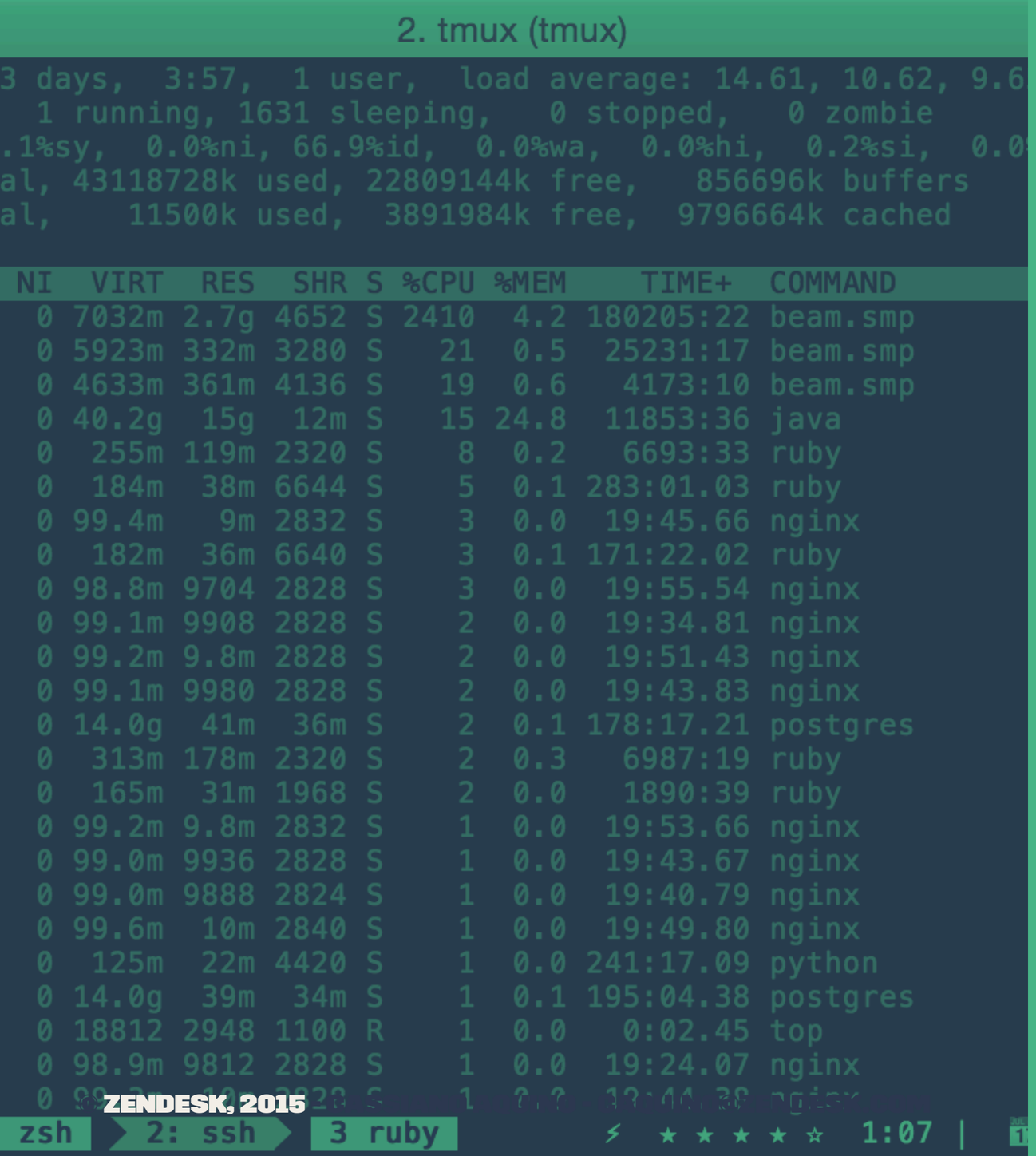
Just a small percentage of the total traffic.
- API calls, Searches, node.save...


CAUTION
EPIC
FAILURE

MONITORING

Important
- Clients health 👀

HOW?
- Report/Exception handlers
- knife status



BUT WHY EVERYTHING IS FAILING NOW?

Important

- erchef ❤ CPU

WHY? WHY?

- Nodes stored on postgresql
- Compressed JSON attributes
- SOLR is for searches only

HOW DO I SPEED UP THINGS?

Old dog, new tricks
Attributes
- Cleanup attributes
- whitelist-node-attrs

Convergence
- Adjust convergence interval
- splay is your friend! 👍

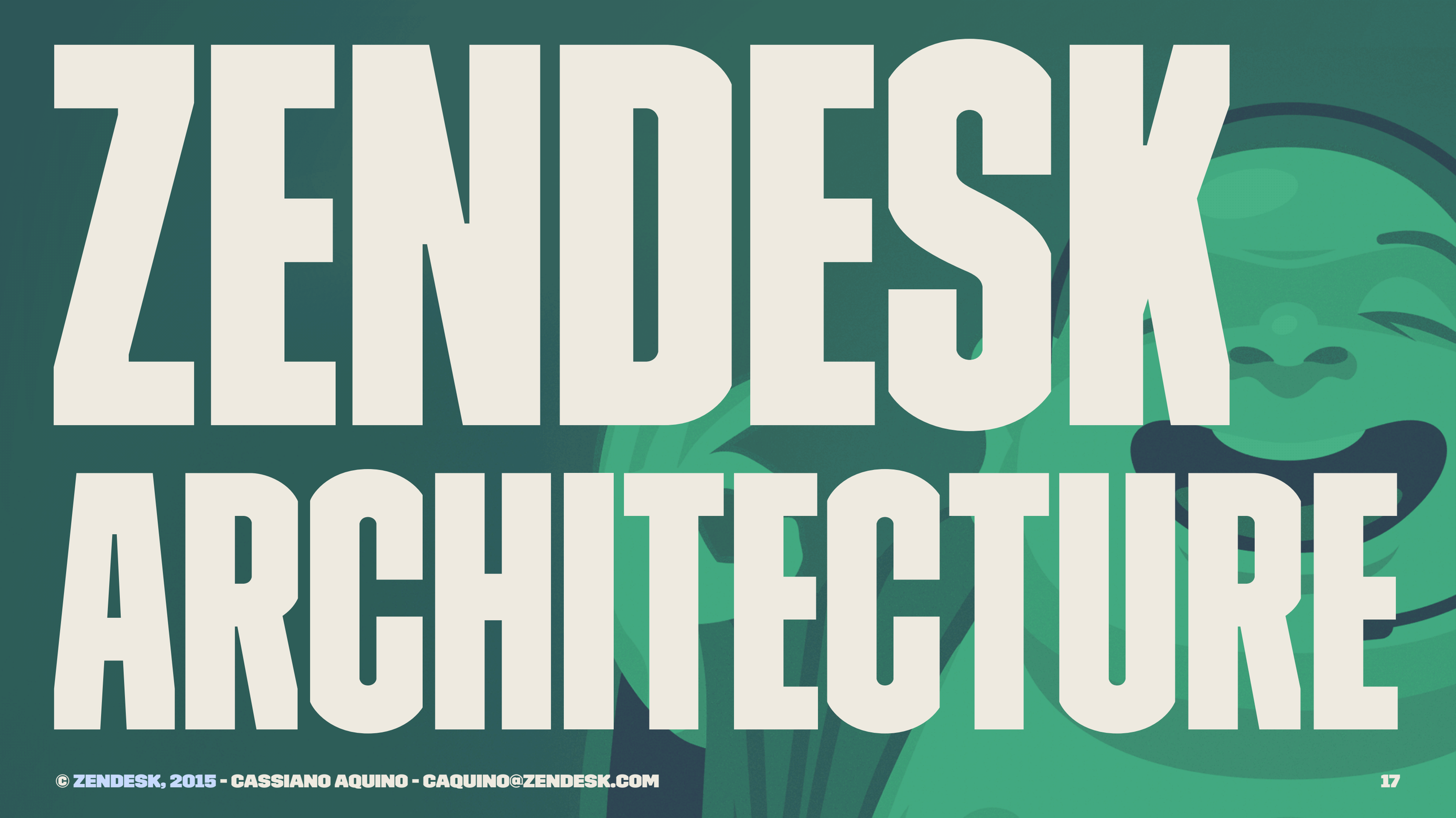
Caching
- node.run_state

CACHING

node.run_state example

```
def find_lb_nodes(service_name)
    nodes = node.run_state["lb_nodes_cache"] ||=
              search(:node, "lb_service:yes")
   nodes.find {|n| n["lb_service_name"] == service_name }
end

web_nodes = find_lb_nodes("http")
db_nodes  = find_lb_nodes("db")
```

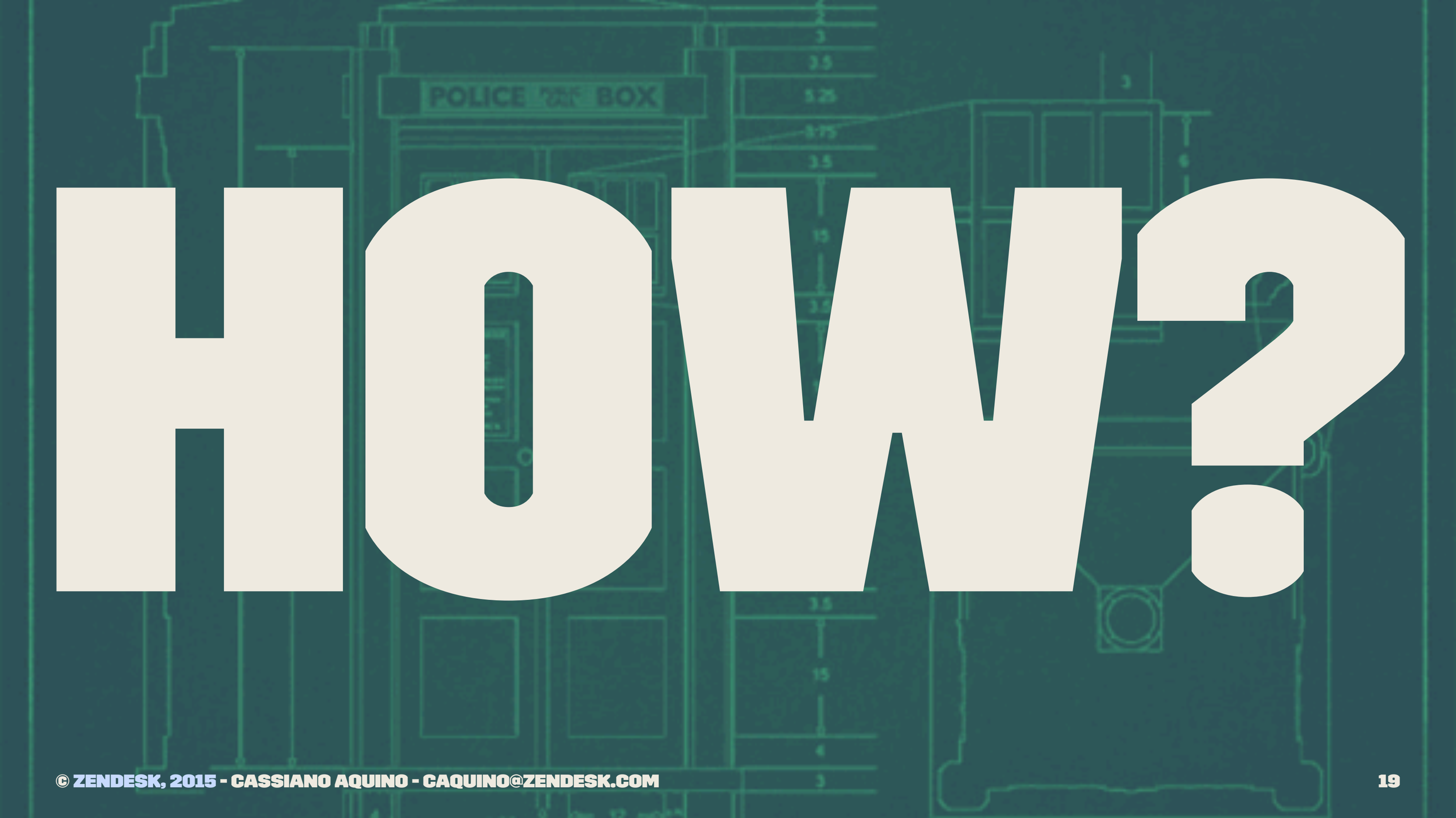
ZENDESK ARCHITECTURE



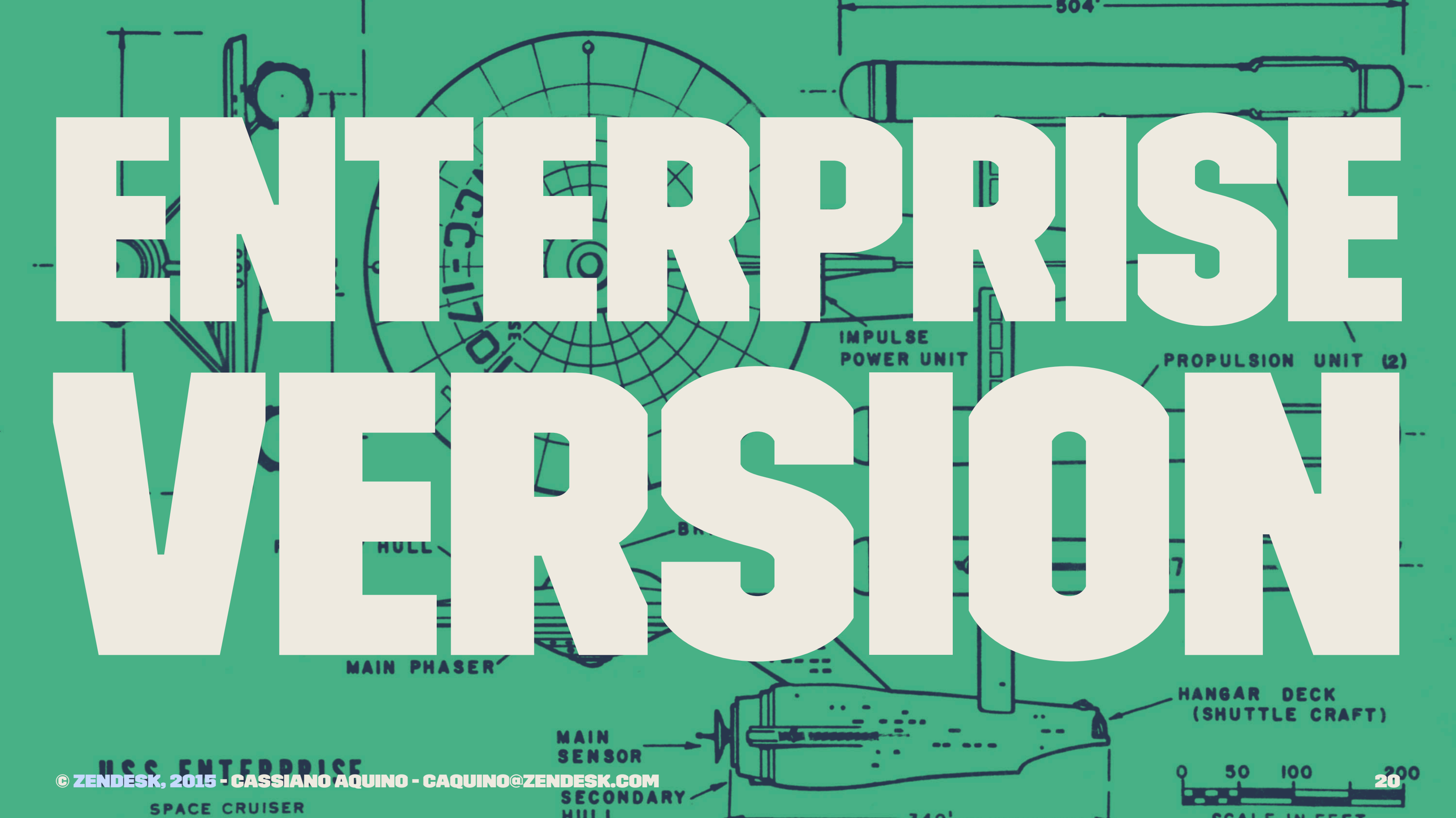
STAGES

Open Source
- Scale up
- Scale out

Enterprise
- HA
- HA + Scale out
- HA + DR + Scale out



HOW?



ENTERPRISE VERSION

TOPOLOGY

Configurations
- HA
- tier
- manual

BACKENDS

```
topology "ha"

server "backend1.domain.com",
  :ipaddress => "192.168.0.22",
  :role => "backend",
  :bootstrap => true,
  :cluster_ipaddress => "194.168.1.22"

server "backend2.domain.com",
  :ipaddress => "192.168.0.23",
  :role => "backend",
  :cluster_ipaddress => "192.168.1.23"

backend_vip "backendvip.domain.com",
  :ipaddress => "192.168.0.24",
  :device => "bond0"
```

FRONTENDS

```
server "frontend1.domain.com",
  :ipaddress => "192.168.0.22",
  :role => "frontend"

server "frontend2.domain.com",
  :ipaddress => "192.168.0.23",
  :role => "frontend"
```

APPLYING THE CONFIGURATION BACKEND AND DRBD

```
on backend1.domain.com
- chef-server-ctl reconfigure

manually configure DRBD on both backends

on backend1.domain.com
- chef-server-ctl reconfigure
```

APPLYING THE CONFIGURATION FRONTENDS

```
from backend1.domain.com to all servers
- copy /etc/opscode

on the remaining servers
- chef-server-ctl reconfigure
```

DR SOLUTION
CHEF-SYNC

Asynchronous distribution
- cookbooks
- enviroments
- roles
- data bags


HELLO
MY NAME IS
Open Source

OPEN SOURCE

backend

```
postgresql['md5_auth_cidr_addresses'] =  [ "127.0.0.1/32",
                                           "::1/128",
                                           "192.168.0.0/24" ]
```

OPEN SOURCE

frontend disable services

```
bootstrap["enable"] = false
rabbitmq["enable"] = false
chef_solr["enable"] = false
chef_expander["enable"] = false
bookshelf["enable"] = false
chef_server_webui["enable"] = false
postgresql["enable"] = false
```

OPEN SOURCE

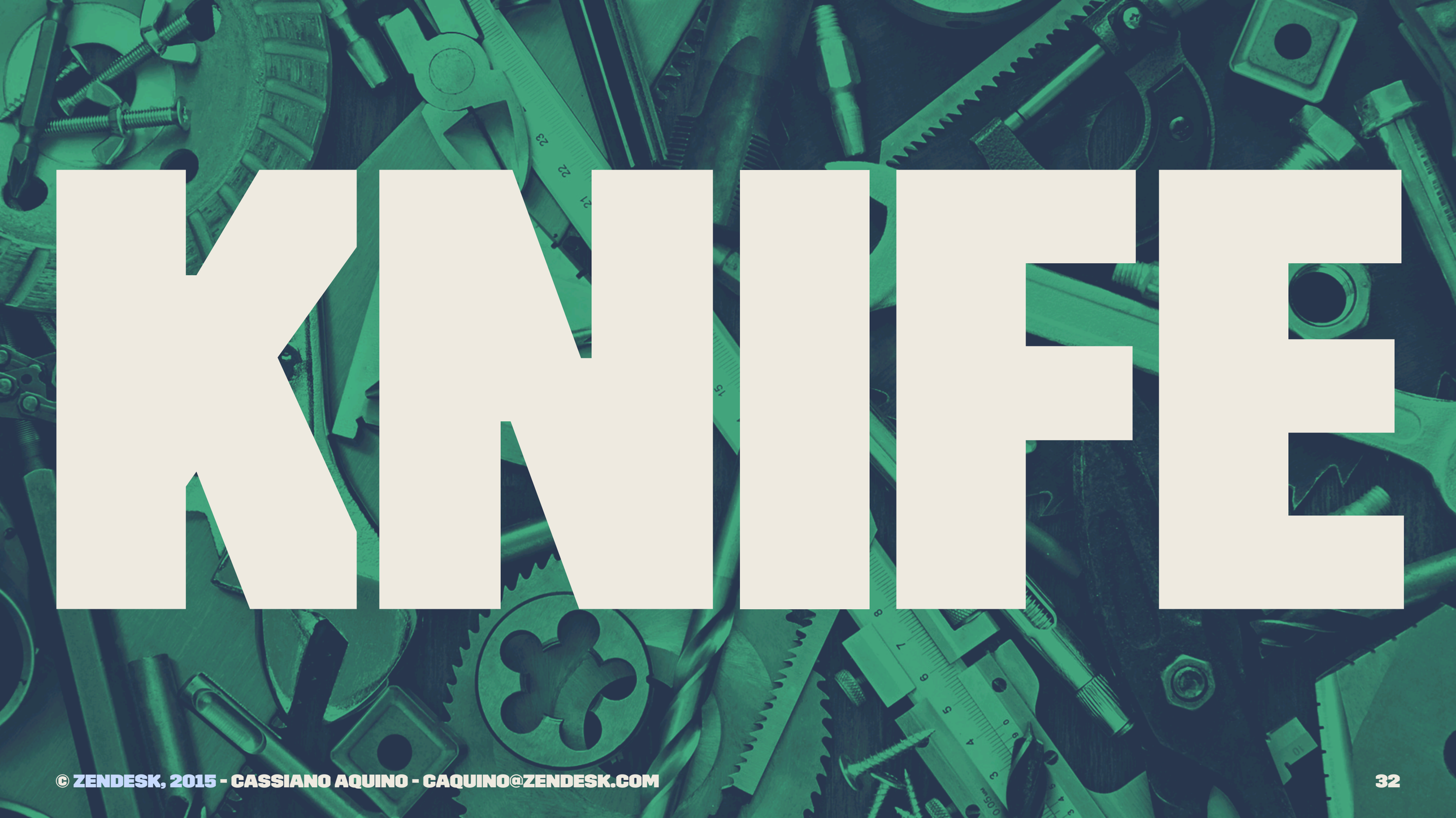
```
use backend services

rabbitmq["vip"] = "192.168.0.8"
rabbitmq["node_ip_address"] = "192.168.0.8"
chef_solr["vip"] = "192.168.0.8"
bookshelf["vip"] = "192.168.0.8"
postgresql["vip"] = "192.168.0.8"
lb["upstream"] = {
        "erchef" => [ "127.0.0.1" ],
        "chef-server-webui" => [ "192.168.0.8" ],
        "bookshelf" => [ "192.168.0.8" ]
      }
```

OPEN SOURCE

small patch
/opt/chef-server/embedded/cookbooks/chef-server/templates/default/sv-erchef-run.erb

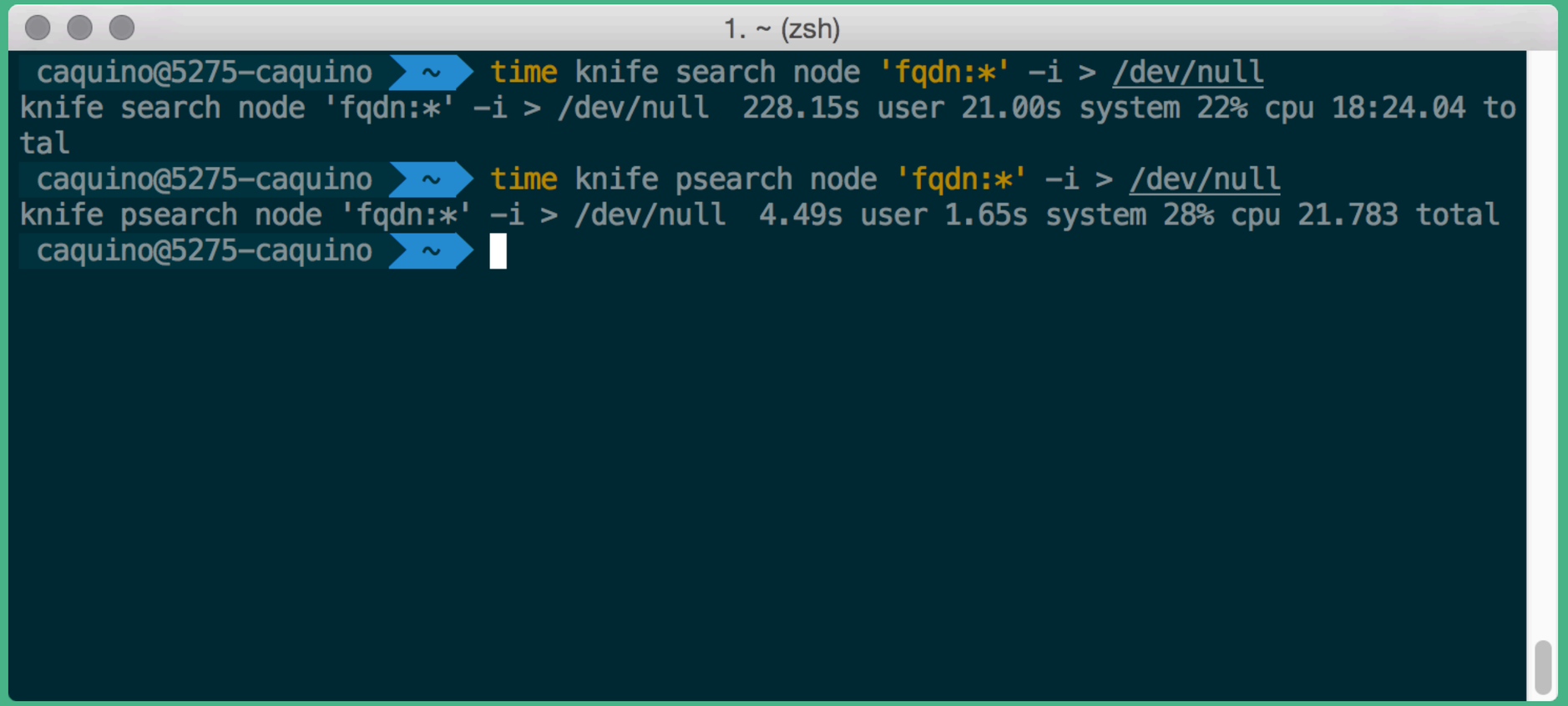
comment out

/opt/chef-server/bin/wait-for-rabbit



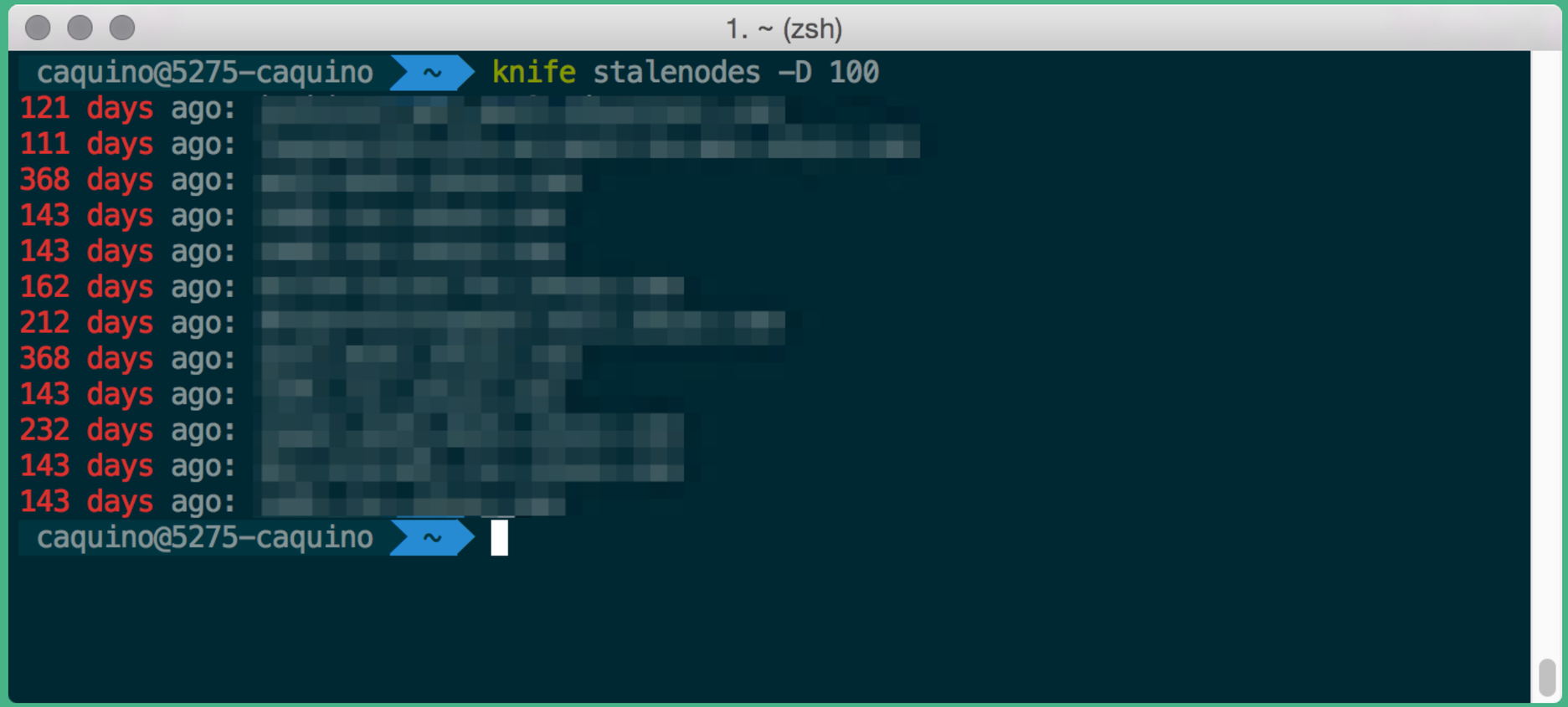
PSEARCH

gem install knife-psearch



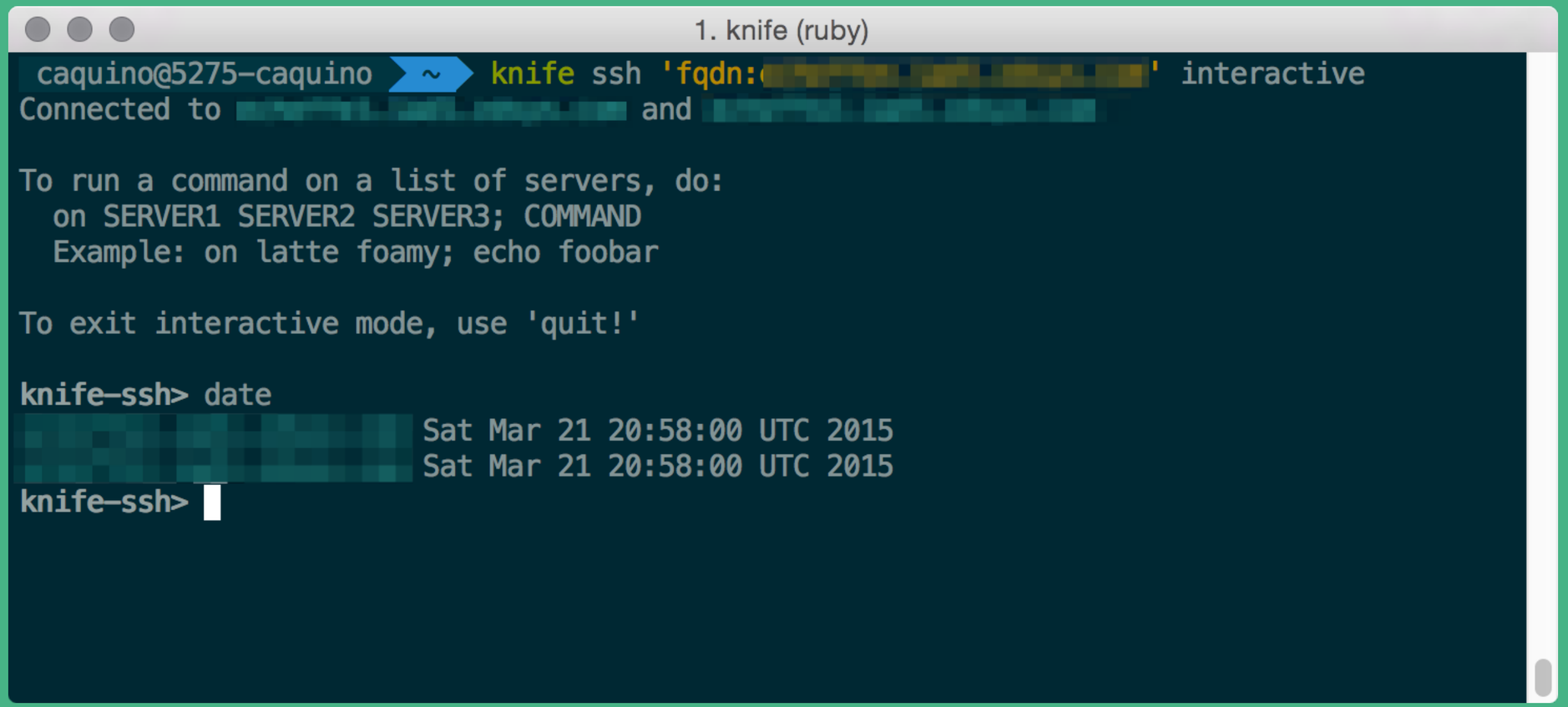
STALE NODES

```
gem install knife-stalenodes
```



KNIFE SSH INTERACTIVE

```
knife ssh 'fqdn:*.domain.com' interactive
```



SEARCH

All about attributes
- knife psearch node "(NOT kernel_machine:x86_64)"
- knife psearch node "virtualization_role:guest"

Range
- knife psearch node "zendesk_config_pod:[4 TO 6]"

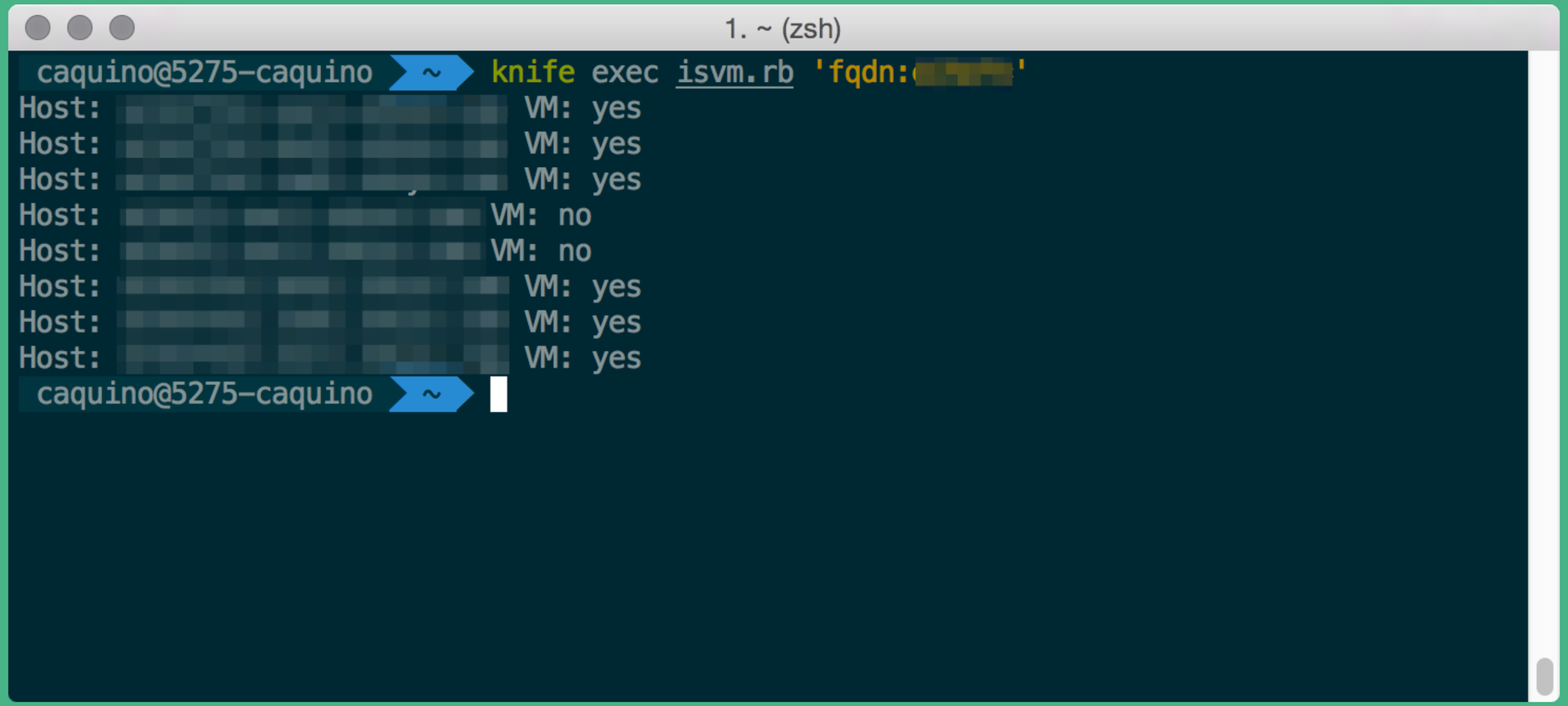
Get to know your attributes
- knife node show -l node.domain.com

KNIFE EXEC SCRIPT

```
Chef::Log.level = :fatal
search(:node, ARGV[2]).each do |node|
  vm = (node["virtualization"] &&
        node["virtualization"]["role"] == "guest") ? 'yes' : 'no'
  puts "Host: #{node['fqdn']} VM: #{vm}"
end
exit 0
```

KNIFE EXEC

```
knife exec is_vm.rb 'fqdn:*.mydomain.com'
```



QUESTIONS?

THANKS

MADE WITH DECKSET